state, private mental attitudes are not. The *conversation state* must include the agents that are assigned to each role, the expressed mental attitudes of participating agents, the conversation protocol in use and the entire history of speech acts sent in this conversation.

The framework considers the effect of context on the meaning of a speech act. Four crucial aspects of the context are identified here:

1. The protocol being followed in the conversation
2. The relationship between the received speech act and the remainder of the discourse
3. The status or authority of the participants (roles)
4. The domain in which the conversation takes place

The first three are stored in the *conversation state*, the fourth is handled by defining a set of rules which must be observed by all agents in a domain. Separating these aspects of context allows more flexible speech acts and protocols to be designed, rather than hardwiring all meaning into a speech act usable in only one context.

We use the following functions in ACL specification:

1. The converse function gives the permissions and obligations ($\in$social_relations) for future speech acts for a certain role ($\in$Roles), given that a certain *conversation state* ($\in$StateVars) holds.
2. Where there is a choice of actions, the select function decides which speech act ($\in$speech_acts) to perform, its decision making is implemented when the protocol is instantiated and is not a part of the protocol specification.
3. Similarly, the add function is not specified as part of the protocol. Its purpose is to update the agent's internal information state ($\in$info_states).
4. The perf_semantic function gives the semantics relating to a performative regardless of what protocol is in use. Given a speech act and state, it returns the new state.
5. The prot_semantic function gives the semantics of the act in the context of this protocol.

converse: Roles $\times$ StateVars $\rightarrow$ social_relations
select: social_relations $\times$ info_states $\rightarrow$ speech_acts
add: speech_acts $\times$ StateVars $\times$ info_states $\rightarrow$ info_states
perf_semantic: speech_acts $\times$ StateVars $\rightarrow$ StateVars
prot_semantic: speech_acts $\times$ StateVars $\rightarrow$ StateVars

sa= <s, perf(r, (C, L, O, i, t_s))>	sa is a speech act from sender s to receiver r with content C, language L, ontology O, identifier i and timestamp t_s
$\|\|$sa$\|\|$ =	the meaning of sa is:
cstate = cstate $\oplus$ (perf_semantic(sa) $\cup$ prot_semantic(sa, cstate))	the sender and receiver(s) must update their conversational state
$\wedge$ $\Delta_{r@tnow}$ = add_r(sa, cstate, $\Delta_{r@tr}$)	the receiver adds content to its state
$\wedge$ converse (role(s), cstate)	the next action must be chosen by the sender
$\wedge$ converse (role(r), cstate)	and receiver

4. Evaluation and Conclusions

The following are some of the requirements for the specification of a protocol, along with comments on how the specification presented above satisfies them:

- *formal and intuitive semantics.* This is provided by the five functions for ACL specification, which can be formally specified using a BDI logic.
- *allowing a precise definition.* Specifications can be made as precise as needed by adding more detailed constraints, an instantiation can be precisely specified by defining the decision making for all functions in the protocol, i.e. by specifying exactly how add and select functions operate.
- *different agent roles within a multi agent system.* Roles are handled by the framework, it is possible to refer to them explicitly in constraints also, or to change roles during the course of a protocol.
- *defining parallelism and choices between different messages.* This is handled by the converse and select functions.
- *(conditional) nesting of protocols in order to perform an intermediate protocol with other agents.* A nested protocol is a handled as a new separate conversation, since all speech acts carry a conversation identifier, messages belonging to the nested protocol can be distinguished from those of the original protocol. The parent protocol will specify (in its converse function) the speech act which initiates the child protocol. When the nested protocol is complete the original will resume, with its original identifier.
- *recursion and repetition on subprotocols are a necessary requirement.* This can be handled by setting up (in the parent conversation) a conversation state variable for the number of desired repetitions, the subprotocol is repeatedly called and the variable decremented (while the value is non-zero).
- *generic interaction protocols which can be instantiated for specific applications.* This is a key feature of the method of specification above which allows generic protocols to be specified, the add and select functions leave open the application specific decision making

Future work will involve the development of a software tool to aid the agent designer in specifying protocols, and managing libraries of subprotocols.

5. References

[1] FIPA, FIPA 97 Specification Part 2: Agent Communication Language. FIPA http://www.fipa.org/spec/FIPA97.html, Nov. 1997.

[2] Pitt, J. & Mamdani, E., A Protocol-Based Semantics for an Agent Communication Language, IJCAI '99. 1999.

A Semantic Framework for Specifying Agent Communication Languages

Frank Guerin and Jeremy Pitt
Intelligent & Interactive Systems, Department of Electrical & Electronic Engineering,
Imperial College of Science, Technology & Medicine, Exhibition Road, London, SW7 2BT
e-mail: {f.guerin,j.pitt}@ic.ac.uk http://www.iis.ee.ic.ac.uk/

Abstract

This paper presents a hybrid semantic framework for the specification of agent communication languages (ACL), unifying intentional semantics with protocol based approaches to provide a more expressive framework. A richer state description based on publicly expressed mental attitudes is introduced. The state description also allows agents to be assigned certain roles in a protocol. Together, this moves to a more workflow-oriented approach, in which conversations are guided by protocols but the effect of individual communicative acts is highly sensitive to the context in which it occurs. We evaluate the usefulness of this framework as a standard for ACLs.

1. Introduction

We aim to provide a general semantic framework within which any ACL can be formally specified. Communication requirements of a typical multi-agent system include: multiple parallel participants, different agentro les within a conversation, consideration of the context in which communication occurs, composition and nesting of protocols and the definition of constraints on message sending. There must also be a systematic, efficient method of development and the possibility of verifying compliance.

The possibility of agents being assigned roles in a conversation and the importance of context point to the need for a rich description of the *conversation state*. Speech act semantics must describe a change in this state. A systematic development methodology is well catered for by allowing protocols to be designed for useful conversations that occur frequently.

We are concerned with unifying the two approaches above, so that protocols can be specified formally without ignoring the meaning of the individual speech acts.

2. Communication Language Semantics

Existing ACLs can be divided into two categories:

1. Protocol based semantics[2]*:* This involves analysing useful conversations in agent interactions and specifying certain protocols for them (with a finite state diagram). The speech acts' meaning is specified in terms of the responses that are allowed at each stage of a conversation.

2. Intentional semantics: These ACLs focus on specifying the meaning of each communicative act in terms of the mentalistic attitudes (beliefs, desires and intentions) of the participants. These specifications give a list of possible speech acts and specify preconditions (mental states that must exist in an agent before transmission) and intended postconditions (states that are desired after) for each speech act. e.g. FIPA's ACL [1].

There are difficulties with each approach. In protocol based approaches, speech acts tend to become meaningless tokens, that have a particular ordering. Intentional semantics do have meaningful speech acts, but often these meanings are specified in terms of the agents' internal mental states which may not be easily accessible, hence an agent's compliance with the language is not easily verified. Furthermore, rigid constraints on the performance of an act limit an agent's autonomy excessively.

We propose a hybrid semantics where speech acts are given a meaning in terms of the publicly expressed mental attitudes rather than the agents' private mental states. We also define protocol semantics, so that speech acts made in the context of a certain protocol commit the conversation's participants to selecting from a possible set of future speech acts. In our framework there is a relation between the protocol semantics and the semantics of individual acts. The obligations that a protocol places on subsequent communications are typically defined in terms of the mental attitudes expressed by the individual acts.

3. The Semantic Framework

We distinguish between an agent's publicly expressed mental attitudes and its personal internal mental attitudes. These need not be identical, agents may not be sincere. Expressed mental states can be easily verified, since they are observable. We treat the conversation as a public construct, all participants are contributing to the *conversation state* with each speech act. Publicly expressed attitudes and information are entered in the